

Learning and Reasoning in Complex Coalition Information Environments: a Critical Analysis

Federico Cerutti ^{*}, Moustafa Alzantot [†], Tianwei Xing [†], Daniel Harborne ^{*}, Jonathan Z. Bakdash [‡],
Dave Braines [§], Supriyo Chakraborty [¶], Lance Kaplan [‡], Angelika Kimmig ^{*}, Alun Preece ^{*},

Ramya Raghavendra [¶], Murat Şensoy ^{||} and Mani Srivastava [†]

^{*} Cardiff University, UK; [†] UCLA, USA; [‡] ARL, USA; [§] IBM, UK; [¶] IBM, USA; ^{||} Ozyegin University, Turkey

Abstract—In this paper we provide a critical analysis with metrics that will inform guidelines for designing distributed systems for Collective Situational Understanding (CSU). CSU requires both collective insight—i.e., accurate and deep understanding of a situation derived from uncertain and often sparse data and collective foresight—i.e., the ability to predict what will happen in the future. When it comes to complex scenarios, the need for a distributed CSU naturally emerges, as a single monolithic approach not only is unfeasible: it is also undesirable. We therefore propose a principled, critical analysis of AI techniques that can support specific tasks for CSU to derive guidelines for designing distributed systems for CSU.

Index Terms—collective situational understanding; artificial intelligence for situational understanding; critical analysis of artificial intelligence techniques

I. INTRODUCTION

Situational understanding requires both insight and foresight. In its traditional definition [17] it is the “product of applying analysis and judgement to the unit’s situation awareness to determine the relationships of the factors present, and form logical conclusions concerning threats to the mission accomplishment, opportunities for mission accomplishment, and gaps in information.” The UK Ministry of Defence Doctrine [1] goes beyond and explicitly mention that (situational) “Understanding involves acquiring and developing knowledge to a level that enables us to know why something has happened or is happening (insight) and be able to identify and anticipate what may happen (foresight).”

Artificial Intelligence (AI) holds the promise to provide efficient and effective methods for supporting humans in situational understanding in a human/machine collaborative effort. When it comes to complex scenarios, the need for a distributed collective situational understanding naturally emerges, as a single monolithic approach not only is infeasible—as argued in [30]: it is also undesirable. Indeed, applying the knowledge representation hypothesis—i.e. that a mechanical embodiment of an intelligent process will appear to have an understanding of the process it encompassess, and its behaviour can be

expressed in casual terms [39]—as a design principle for mechanical embodiment of intelligent processes naturally leads to focus on fully qualified causal knowledge-based systems. Although such an assumption is not necessary for building a mechanical embodiment of an intelligent process, it does suggest that human expectations and understanding of the mechanical embodiment would search for a (propositional) account of the knowledge and causality of the decision. In the following, we will assume that this is the case, hence it is undesirable to provide decision makers with mechanical support for which the human cannot identify elements of knowledge and causality.

Unfortunately, to the best of our engineering abilities and independent of the large variety of techniques we can employ, these systems will always suffer from at least two problems [27]: 1) we cannot list all the preconditions for an action—e.g., switching on a combustion car engine requires there to be no potatoes in the exhaust tube—also known as the *qualification problem*; and 2) we cannot envisage all the effects for an action—sometimes referred in popular literature as *butterfly effect*—also known as the *ramification problem*. This leads to the need for very specific systems, so specific that the risks posed by these two problems become—if not negligible—acceptable. Hence, as also argued in [16], highly engineered task specific machinery can collaborate to achieve more complex tasks. But, since the generality of many approaches developed in AI, the question of selecting such task specific machinery arises.

We propose a principled, critical analysis of AI techniques that—when they have not been already put in use—at least in principle can support specific tasks of interest for the insight and foresight aspects of situational understanding. This is clearly not an entirely novel idea, and in Section II we review some of the existing work in the area, as well as discussing the motivations for this work with specific details of the tasks we focus on. The novelty of this work relies on its purpose of seeing the metrics that compose our critical analysis—Section IV-A—as guidelines for designing distributed systems for collective situational understanding. This will then be exemplified with a case study in Section V.

II. MOTIVATION AND BACKGROUND

A. Motivation

As a motivating scenario, we will consider the case of soldiers carrying out reconnaissance in a contested urban

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

environment. They are supported by drones equipped with camera and GPUs for image processing *in loco* as well as CPU power for (limited) threat assessment.

Kinetic actions will increasingly take place “in urban environments, if only because by 2040 two-thirds of the world’s population will be living in cities. [...] Intense urban warfare, as demonstrated by the recent battles for Aleppo and Mosul, remains grinding and indiscriminate, and will continue to present difficult problems” [3].

Moreover, “many of the emerging technologies will also be available to [...] adversaries” [3] and this includes autonomous weapons. Under present international law, the use of autonomous weapons is forbidden [12]: there is the requirement to keep a human in the loop—i.e., “with a human constantly monitoring the operation and remaining in charge of critical decisions” [4]. However, there is no guarantee that adversaries will do the same. “They might, for example, decide on pre-delegated decision-making at hyper-speed if their command-and-control nodes are attacked” [2], thus acting with humans out of the loop, “with the machine carrying out the mission without any human intervention” [4].

The varying amounts of human control over machine functions are called Levels of Automation (LOAs) [41]. Higher levels of automation are best suited for well-specified rule and skill-based tasks with low uncertainty—e.g., calculating the physics of the firing solution for artillery based on the location of prior shots, distance, elevation, weather, and weapon capabilities (but not firing the actual weapon), whereas more human control is best suited for high uncertainty tasks requiring knowledge and expertise—e.g., Captain Sullenberger’s manual landing of Flight 1549 on the Hudson River following a complete engine failure [11].

In contrast to LOAs, which strictly separate the allocation of functions between humans and machines, a more modern approach is that humans and machines work together collaboratively [11]. In a coalition context, collaboration among partners is essential. Human-machine collaboration, or human-agent teaming, in tasks such as weather forecasting and chess have demonstrated that teaming leads to better performance than either humans alone or machines alone [38]. Developing broader applications for human-agent teaming in coalitions requires advances in AI for learning and reasoning. Such advances include: Mechanisms for facilitating collaborative interactions (between and within coalitions of humans and machines), techniques uncertainty quantification and causal reasoning about the uncertainty, and effectively representing the uncertainty and causal reasoning from machines-to-humans and vice-versa. Although human-agent teaming allows for more flexibility than LOAs, hard limits on what agents or machines can perform on their own (e.g., not firing a lethal weapon without a human decision) are still paramount in safety-critical environments.

B. Background in Critical Analysis of Machine Learning Approaches

Our focus is on contested urban environment with dispersed team of humans and machines accessing heterogeneous information sources, with the need for learning in new environments in presence of persistent threats.¹

While we will not comment in this paper on how humans learn in new environments, in the presence of large volume of data machine learning systems are commonly employed to generate predictive models. A computational system is said to *learn* from experience, with respect to some class of tasks and performance measure P , if its performance at tasks, as measured by P , improves with experience [28]. Machine learning systems generate predictive models on the basis of experience gained by analysing training instances described through observed characteristics or *features*. An—often unstated—assumption here is that the set of training instances is representative of the population on which the trained model will be exploited. When this assumption does not hold, it might be that there are (undesirable) biases in the training set leading to (undesirable) inferencing. Given the complexity of the issue, in this paper we will not discuss it further, but an interested reader is referred to [10].

Works analysing the landscape of existing learning and reasoning techniques focus on limited dimensions and they provide little guidance on how to choose—let alone how to compose—approaches for a given task. To cite a recent example, at the IJCAI 2016 Workshop on Deep Learning for Artificial Intelligence, David Gunning² illustrates an analysis of the current machine learning techniques based on three dimensions: 1) the assumed model underlying the machine learning process; 2) their average accuracy in common scenarios; 3) their level of explainability. While the first dimension is undebatable—i.e., deep learning approaches are based on neural networks, Bayesian networks are graphical models, etc—the other two deserve additional comments.

The first comment is that accuracy is domain and task dependent. For instance, in [34] the authors summarise three years of improvements in computer vision tasks, to illustrate the increase of accuracy of using deep neural networks for computer vision. However, those results vary significantly on the basis of the given task. For instance, while the image classification task has seen an improvement from 2012 (16% error) to 2014 (6%), localising a single-object in 2014 still carried a 25% error.

The second comment is that explainability groups together a large variety of dimensions—already explored in other papers such as [9]—that would distract from the main contribution of this paper. For instance in [9] the authors summarise the main dimensions for interpretability, such as model transparency, simulability, decomposability, and algorithm transparency; and

¹This aligns with the research context highlighted by Dr. Tien Pham of the US Army Research Laboratory at the DAIS ITA Annual Fall Meeting, 27th September 2017, Slides available at <https://goo.gl/n7VvP8> (on 8 March 2018).

²<https://goo.gl/FJZ96Q>, page 4 (on 8 March 2018).

model functionality, in terms of textual description, visualisation, and local explanation.

III. ARTIFICIAL INTELLIGENCE TOOLS CONSIDERED

In the following we focus on four machine learning approaches representative of a larger set of state-of-the-art, widely used, methods: convolutional neural networks; probabilistic graphical models; probabilistic logic programming; and topic modelling.

A. Convolutional Neural Networks

Convolutional neural networks (CNNs) nowadays represent state of the art methods for many tasks in computer vision. They have been successfully used to deliver outstanding results in image recognition, object localization and detection, semantic segmentation, and other tasks. CNNs are a special kind of feed-forward neural network that was first proposed by Yann LeCun in the early 1990s [25]. They gained considerable popularity in the last decade after they were used by all winning teams in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competition since 2012 [24].

CNNs are distinguished from other feed-forward neural networks by having groups of *convolution* and *pooling* layers followed by fully connected layers. The *convolution* layers each will have a set of learnable filters and convolution output is produced by the dot product of the filter with small regions of the input image volume in a sliding window fashion. This way, the same set of weights (i.e., the filter) is re-used when computing the filter output for different parts of the image. In addition to convolution layers, the pooling layers are often used to reduce the size of convolution outputs to achieve computation efficiency and provide spatial invariance when detecting objects in the image. CNNs are trained using gradient descent with backpropagation and the speed of training can be dramatically accelerated by using GPUs for parallel processing [6].

In addition to achieving super-human accuracy on image recognition, CNNs have been also successfully used in other tasks. For example, Regional-CNN (R-CNN) [19] is now the state-of-the-art method for object detection. Compared to traditional CNNs, R-CNN can detect multiple objects in the same image as well as outputting the bounding box of each predicted object. Recent enhancement to reduce the computation costs of R-CNN have been proposed including: Faster-CNN [32] and YOLO [31]. Other extensions such as Mask R-CNN [20] have also been proposed to higher quality boundaries of detected objects by computing pixel wise segmentation of the input image.

B. Probabilistic Graphical Models

Probabilistic graphical models provide the mechanisms to develop reasoning engines for a non-deterministic model of the world. In short, these graphical models simply represent the joint probabilities of a collection of variables and reveal the Markov relationships between the variables as a network in graphical form. For Markov networks, the graphs are

undirected, and for Bayesian networks, they are directed [23]. The actual manner of how variables interact in a Markov network is revealed through factor graphs that encode the joint probabilities as a collection of potentials encapsulating a subset of dependent variables. Conditional probabilities encode the overall joint probabilities in a Bayesian network. Exact inference methods to determine the probability of latent variables conditioned on the observed values exist, but in general are not scalable. Exact efficient inference methods do exist when the networks exhibit a tree structure, but approximate methods must be employed when the graphs includes loops [40].

Graphical models have been extended to capture external knowledge. Markov logic networks incorporate first order logic through a set of soft logic rules whose probabilistic enforcement strengths are learned over training data [33]. The rules build up a Markov network for reasoning over variables that connect to various class types. Similarly, multiple-entity Bayesian networks uses a series of rule-based fragments to compose Bayesian network for reasoning [42].

The distributions of the parameters for graphical models are usually accounted for during the training process, but the parameters are typically treated as known constants during inference. There are efforts to account for uncertainty in the inferred probabilities due to parameter uncertainty from limited data, which is a problem in training up a system in the contested environment described earlier, e.g. [29]. In [8], we have compared the uncertainty characterization capabilities of such networks using world financial market data. In short, subjective Bayesian networks augment the inference capability of Bayesian network with an uncertainty value that enables one to set up confidence bounds at any significance level. Future work is needed to consider uncertain inference over logical models such as Markov logic networks.

C. ProbLog

ProbLog [15], [18]³ belongs to a family of probabilistic logic programming (PLP) languages [14] following Sato's distribution semantics [35]. It extends logic programming by annotating some ground facts with their probability of being true, which generalizes a single program into a distribution over programs that share their rules, but differ in their databases. More specifically, a ProbLog program consists of two parts, a set F of ground probabilistic facts $p : : f$ where p is a probability and f a ground atom, and a set R of rules $h :- b_1, \dots, b_n$ where h is a logical atom and the b_i are literals.⁴ While the semantics is defined for countably infinite sets of probabilistic facts, see [35] for details, we restrict the discussion to the finite case in the following. ProbLog considers the ground probabilistic facts as independent random

³More information on ProbLog, including an open source implementation and an interactive online tutorial, can be found at <https://dtai.cs.kuleuven.be/problog/>.

⁴For the semantics of ProbLog to be well-defined, the set of rules has to have a two-valued well-founded model for each subset of the probabilistic facts: a sufficient condition for this is for programs to be stratified, i.e., have no loops through negation. See [14], [18] for further details.

variables, i.e., we obtain the following probability distribution P_F over truth value assignments to sets of ground facts $F' \subseteq F$: $P_F(F') = \prod_{f_i \in F'} p_i \cdot \prod_{f_i \in F \setminus F'} (1 - p_i)$. As each logic program obtained by choosing a truth value for every probabilistic fact has a unique least Herbrand model, P_F can be used to define the *success probability* $P(q)$ of a query q , that is, the probability that q is true in a randomly chosen such program, as the sum over all programs that entail q : $P(q) := \sum_{\substack{F' \subseteq F \\ \exists \theta F' \cup R \models q\theta}} P_F(F') = \sum_{\substack{F' \subseteq F \\ \exists \theta F' \cup R \models q\theta}} \prod_{f_i \in F'} p_i \cdot \prod_{f_i \in F \setminus F'} (1 - p_i)$.

Inference in ProbLog is concerned with computing marginal probabilities of queries, i.e., ground atoms, under this distribution, potentially conditioned on a conjunction of evidence atoms. While this is a #P-hard problem in general, ProbLog relies on state-of-the-art knowledge compilation techniques to achieve scalable inference across a wide range of models.

The parameters of ProbLog programs can be learned from partial interpretations [18], and ProbLog rules defining a target predicate can be learned from a ProbLog program specifying background knowledge (in the form of facts and/or known rules for other predicates) and ground atoms using the target predicate annotated with target probabilities [13].

D. Topic Modelling using LDA

Topic Modelling is a form of machine learning in which a statistical model is created to learn about topics that are present in a series of articles or documents. A form of topic modelling is Latent Dirichlet Allocation, or LDA, [7], a three-level hierarchical Bayesian model,⁵ in which each item of a collection is modelled as a finite mixture over an underlying set of topics. Each topic is, in turn, modelled as an infinite mixture over an underlying set of topic probabilities. In the context of text modelling, the topic probabilities provide an explicit representation of a document. Although more advanced techniques have been recently proposed, e.g., [43], in the following we will focus on the original LDA proposal as an example of topic modelling.

IV. CRITICAL ANALYSIS

To achieve our overall goal to derive guidelines for designing distributed systems for collective situational understanding, we first need to introduce useful metrics.

A. Metrics

We identify classification metrics belonging to three main classes:

- 1) Structural properties;
- 2) Economic properties;
- 3) Quality assurance properties.

⁵LDA is in essence an instance of a probabilistic graphical model, but we will treat it here separately as it represents a case of unsupervised machine learning system.

1) *Structural properties*: Machine learning algorithms can be classified on the basis of their *learning typology* as well as of their *artificial society typology*.

Currently one of the most popular typologies is *supervised* learning where, given a set of inputs \vec{X} and of outputs \vec{Y} (labels), we assume that there exists a function $f : \vec{X} \rightarrow \vec{Y}$ and the goal of the learning machinery is to approximate f on the basis of the given data. *Unsupervised* learning instead receives a set of inputs \vec{X} but neither targets outputs nor receives rewards from its environment. The goal then is to identify function(s) $k : \vec{X} \rightarrow \vec{Z}$ for given tasks such as classifying unlabelled data. *Semi-supervised* learning sits between supervised and unsupervised learning, relying on the assumption that unlabelled data is significantly cheaper than labelled data. Other approaches are also possible: e.g., *reinforcement* learning has become particularly popular recently [37], where the learning algorithm interacts with the environment and receives rewards or punishments. Orthogonal to this categorisation is the *feature selection* process, such as whether features are manually identified by engineers or automatically by the learning algorithm.

Approaches to machine learning can leverage different artificial society typologies. Focusing on the training activities, they can be centralised when performed by a single agent, or they can require the contribution of multiple agents. Depending on the characteristics of the multi-agent society, the learning activities can be distributed across a network of peer-to-peer nodes; these nodes may be either indiscernible—i.e., all nodes perform the same exact task and collectively they address a harder task than each individually—or discernible—i.e., nodes can perform specialised tasks. The latter is also pragmatically similar to the case where the multi-agent society identifies hierarchical structures, where naturally different nodes will have different levels of responsibility towards the overall training task. We refer an interested reader to [9] for further details on distributed model learning, especially in the context of neural networks.

2) *Economic properties*: Machine learning algorithms can be classified also on the basis of their *input data efficiency* and *parsimony of the learnt model*.

The quantity of data needed for training by different learning algorithms can vary significantly, hence algorithms can be classified on the basis of whether and how they can address the problem of sparsity of data. Some approaches might allow for tellability—i.e., enabling a human expert to provide specific model parameters—or they might use a model—i.e., a computational effective surrogate of a human expert—to reduce the need for data, or even allow for biases in the model parameters. Such a model will often be based on expert knowledge, hence partially overlapping with tellability.

A clearly-desirable property for each learnt model is that it is minimal, i.e., that it is the most parsimonious model—in terms of the chosen underlying mathematical assumptions—to address the given task without significantly decreasing its quality, as discussed in the following class of properties.

3) *Quality assurance properties*: Like any other software product, the quality of machine learning algorithms can vary on the basis of their *inferencing accuracy*, *confidence representation*, *robustness*, *inferencing efficiency*, and *reasoning capabilities*.

Regarding inferencing accuracy, as we discussed in Section II-B, this can be formalised with a measure of the distance between the learnt model, and the true mathematical function that the model approximates. Therefore it is meaningful only when such mathematical functions can be formalised at least in terms of domain and co-domain.

Uncertainty-aware reasoning systems such as subjective Bayesian networks can provide a confidence interval. The quality of the uncertainty representation can be evaluated over simulations where a ground truth can be established. Then, one measure of the quality is the deviation between the desired significance level of the bound and the actual significance level measured as the fraction of time the ground truth falls within the interval.

Uncertainty awareness is also a desirable property of robust learning systems. As discussed in [16], robustness can manifest against known unknowns, but also unknown unknowns. For the purpose of this paper, we will focus only on robustness against the open set classification problem, i.e., where an algorithm is asked to identify elements upon which it has never been trained.

Once the learnt model is created, a different question is about how efficient it is implemented in a current engineering architecture, i.e., choosing between integer vs fixed point vs floating point numbers can have a significant impact on hardware costs, runtime, and energy efficiency of implemented model.

Moving from the engineering domain into a more theoretical account of the computational complexity of the task, different machine learning approaches can have different levels of expressiveness as well as reasoning capabilities. For this work, we will consider in particular whether the derived system is able to handle symbolic reasoning—and to what complexity.

B. Analysis

Table I summarises an analysis of the four approaches listed in Section III against four of the dimensions highlighted in Section IV-A. Although most of the measures should be made more precise, e.g., adding specific references to the training sets or the specific classes to be considered and measures derived from the confusion matrix, we choose here simply to give a qualitative assessment over a 5-point scale (Very Low, Low, Medium, High, Very High).

As it will become manifest in the subsequent section, the goal of this study is not to perform a classification exercise, rather to develop a critical methodology with the ultimate goal of supporting situational understanding.

V. A CASE STUDY FOR SITUATIONAL UNDERSTANDING

In the operational setting highlighted in Section II, the situational understanding task that we want to perform concerns an

TABLE I
SUMMARY OF EVALUATION OF APPROACHES SUMMARISED IN SECTION III AGAINST FOUR DIMENSION AMONG THOSE HIGHLIGHTED IN SECTION IV-A.

	Input Data Efficiency	Object Detection Accuracy	Robustness w.r.t. Open Set	Symbolic Reasoning (Expressiveness)
CNN	Very Low	Very High	Very Low	Unknown (cf. [36] for a discussion)
BN	Very High	Very Low	Very Low	Yes: analogous to probabilistic inference over a PL KB
ProbLog	Very High	Very Low	Very Low	Yes: analogous to probabilistic inferences over a subset of FOL KB
LDA	Medium	Very Low	High	No

alarm system that should be triggered when detecting a flying drone and a human using a video stream. The overall system must have a certain level of input data efficiency, in particular tellability, as large datasets for training on dangerous situations are unfeasible due to the fact that (hopefully) are quite rare; it should have high accuracy in analysing the video stream, as it is the main sensing modality; it should be as robust as possible to the open set classification problem, as eloquently argued in [16].

From Table I it is evident that singularly none of the approaches can address the given situational understanding task. However, a combination of CNN, BN (or ProbLog), and LDA does. In particular, Bayesian networks allow for tellability from an expert, hence they could potentially provide an interface with humans. CNN have very high object detection accuracy, hence they can be used to feed into one of the random variables of the BN. However, CNN have very low robustness w.r.t. the open set classification problem: we can however use LDA to evaluate the semantic distance between the concepts identified—eventually with low probability—by the CNN and the concepts known by the Bayesian network. This relies upon the assumption that semantically close concepts might share features that could have been relevant in the CNN inferencing step.

A. Tellable Alarm System

Figure 1 depicts a Bayesian network capturing the idea that an alarm system should be triggered when detecting a flying drone: for sake of conciseness we overlook the presence of humans due to space constraints. Following [8], we can assume that the structure of the network has been obtained through an interactive process with a human expert: for instance, there is clearly a statistical dependency between the random variables representing observations of the presence of a drone and of

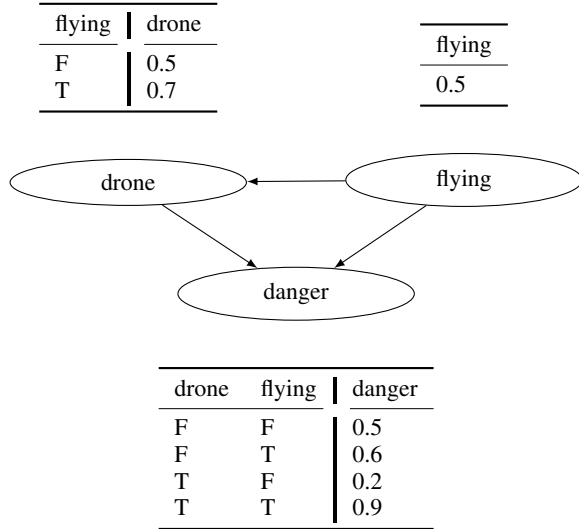


Fig. 1. Bayesian network for the risks associated with a flying object, in particular when it is a drone

the presence of a flying object. However, we can also assume that some of the conditional probabilities are learnt from several hours of patrolling observation. For instance, since those observations took place in a urban environment, most flying objects of interest would probably be drones—instead of helicopters or airplanes—hence why observing a flying object would lead with probability 0.7 to infer that such an object is a drone.

B. Accurate Object Detection

We used the object-detection API [22] in TensorFlow [5] open-source deep learning framework to deploy an instance of the MASK-CNN with ResNet101 [20] model trained on the Common Objects in Contexts (COCO) dataset [26]. This model achieves near state-of-the-art mean average precision score $\text{mAP} = 33\%$ on the COCO test set while producing high quality segmentation of the detected objects.

To illustrate the use of this object detector, let us consider Figure 2⁶ depicting a person and a flying drone. According to our scenario this should trigger an alert, but the output of the object detector, depicted in Figure 3, includes a person with confidence of 99%, an airplane with confidence 34%, and a traffic light with confidence 34%.

C. Robustness against Open Set Classification

A straightforward implementation of LDA executed on the Wikipedia page for *airplane*⁷ would result in the following topics listed in decreasing order of importance: wing; aircraft; plane; flight; engine.⁸ Among the random variables considered

⁶https://pixabay.com/p-499033/?no_redirect on 18 February 2018, released under CC0: public domain.

⁷<https://en.wikipedia.org/wiki/Airplane> (on 1st March 2018).

⁸Running it on the Wikipedia page of traffic light https://en.wikipedia.org/wiki/Traffic_light (on 1st March 2018) does not return useful results, hence we ignore this in the remainder of this paper.



Fig. 2. Image depicting a person and a drone.

in Figure 1, *flying* shares the same stem as one of the top results from the LDA approach, hence it might be sufficient evidence to trigger accordingly the Bayesian network. This results in a probability for *drone* as 0.7 and a probability for *danger* as 0.81.

D. Increasing the Expressivity of the System

From the previous section it becomes evident that enriching the semantics of random variables—hence enabling a certain level of logical reasoning—would be an advantage. From Table I it is evident that ProbLog allows to directly encode Bayesian networks, and to easily integrate them with more expressive models. Our example Bayesian network could be written in ProbLog as follows:

```

0.5::flying.
0.5::drone :- not flying.
0.7::drone :- flying.
0.5::danger :- not drone, not flying.
0.6::danger :- not drone, flying.
0.2::danger :- drone, not flying.
0.9::danger :- drone, flying.
  
```

In ProbLog, we can easily extend such a propositional model to both handle *relational domains*, e.g., involving flexible numbers of objects and dependencies between properties of



Fig. 3. Image depicting a person and a drone, processed by CNN with low accuracy.

different objects, and to probabilistic models that do *not* satisfy the independence assumptions of Bayesian networks. For instance, we could have several independent sensors that determine (with different accuracy) whether something is flying, a relational rule saying that objects close to drones may be drones as well, independently of whether they are flying or not, and the level of danger may increase with the number of objects in each category:

```
% generic model
Accuracy::flying(Y,X) :-
    object(X), sensor(Y,Accuracy).
flying(X) :- flying(_,X).
0.5::drone(X) :- object(X),not flying(X).
0.7::drone(X) :- object(X),flying(X).
0.1::drone(X) :- object(X), object(Y),
    close(X,Y), drone(Y).
0.5::danger :- object(X),
    not drone(X), not flying(X).
0.6::danger :- object(X),
    not drone(X), flying(X).
0.2::danger :- object(X),
    drone(X), not flying(X).
0.9::danger :- object(X),
```

```
drone(X), flying(X).
```

We can then combine this general model with situation-specific input on the sensors, objects and observations of interest:

```
% specific input
sensor(camera1,0.9).
sensor(sensor2,0.6).
object(o1).
object(o2).
object(o3).
close(o1,o2).
close(o2,o1).

% observations
evidence(drone(o2)).
evidence(not(flying(o3))).
evidence(flying(camera1,o1)).
```

Moreover, ProbLog allows us to explicate the reasoning line that emerges from the usage of LDA (cf. Section V-C) in a rather straightforward way, e.g.:

```
0.12::flight(X) :- airplane(X), object(X).
0.05::flight(X) :- flying(X).
0.05::flying(X) :- flight(X).
```

ProbLog's generic inference engine directly handles such more complex models, instantiating them on demand as required, and thus allows the user to focus on high level modelling. As with all expressive languages, the complexity of inference depends on the complexity of the model.

Furthermore, probabilistic rules defining a target predicate can be learned from data [13]. This requires a ProbLog program modelling the available background knowledge, which could be in itself a ProbLog or Prolog model, or simply a database of facts, as well as a set of examples, which consist of ground instances of the target predicate together with their desired probabilities. For instance, a possible target predicate could be `danger(ID)`, where `ID` is a scene identifier, the background knowledge could provide information on which objects exist in the scene, which objects are drones, flying, close to each other, etc, and examples could comprise positive and negative instances of the form `(danger(1),1.0)` and `(danger(2),0.0)` respectively, and examples with intermediate probabilities such as `(danger(3),0.4)`. As usual in inductive logic programming, the space of possible rules has to be specified through mode declarations, which essentially define how different predicates can be joined in the body of a rule.

VI. CONCLUSION

In this paper we discussed a principled, critical analysis of AI techniques that can support specific tasks for CSU to derive guidelines for designing distributed systems for CSU. AI advances in learning and reasoning have enormous potential for facilitating human-agent teaming, including CSU for coalitions. Coalitions require partnerships that are collaborative. We

contend that the aim for AI in coalitions should be effective collaboration among humans and machines, rather than the less flexible approach used in levels of automation which separate the functions of humans and machines.

Although we advocate composition of systems for enhancing overall robustness, it is worth noticing that the case study we highlighted in Section V could be used as a starting point for further approaches to regularise CNN using logical rules, an area first discussed in [21]. Indeed, this paper is just a preliminary exploration providing insights for future, more specific and case-based, critical evaluations of learning and reasoning approaches in complex coalition information environments.

REFERENCES

- [1] Understandig: Joint Doctrine Publication 04 (JDP 04). Ministry of Defence, UK, 2016.
- [2] Autonomous weapons are a game-changer. *The Economist: Special report on the Future of War*, pages 15–16, 27th January 2018.
- [3] Preparing for more urban warfare. *The Economist: Special report on the Future of War*, page 9, 27th January 2018.
- [4] The new battlegrounds. *The Economist: Special report on the Future of War*, pages 3–4, 27th January 2018.
- [5] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.
- [6] T. Akiba, S. Suzuki, and K. Fukuda. Extremely large minibatch sgd: Training resnet-50 on imagenet in 15 minutes. *arXiv preprint arXiv:1711.04325*, 2017.
- [7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [8] D. Braines, A. Thomas, L. Kaplan, M. Şensoy, M. Ivanovska, A. Preece, and F. Cerutti. Human-in-the-loop situational understanding via subjective Bayesian networks. In *Proc. of the 5th International Workshop on Graph Structures for Knowledge Representation and Reasoning*, 2017.
- [9] S. Chakraborty, A. Preece, M. Alzantot, T. Xing, D. Braines, and M. Srivastava. Deep learning for situational understanding. In *2017 20th International Conference on Information Fusion (Fusion)*, pages 1–8, July 2017.
- [10] D. K. Citron and F. A. Pasquale. The Scored Society: Due Process for Automated Predictions. *Washington Law Review*, 89:1–33, 2014.
- [11] M. M. Cummings. Man versus machine or man + machine? *IEEE Intelligent Systems*, 29(5):62–69, 2014.
- [12] N. Davidson. Autonomous weapon systems under international humanitarian law. *UNODA Occasional Papers*, No. 30, Jan. 2018.
- [13] L. De Raedt, A. Dries, I. Thon, G. Van den Broeck, and M. Verbeke. Inducing probabilistic relational rules from probabilistic examples. In *Proceedings of 24th International Joint Conference on Artificial Intelligence*, pages 1835–1842, 2015.
- [14] L. De Raedt and A. Kimmig. Probabilistic (logic) programming concepts. *Machine Learning*, 100(1):5–47, 2015.
- [15] L. De Raedt, A. Kimmig, and H. Toivonen. ProbLog: A probabilistic Prolog and its application in link discovery. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2462–2467, 2007.
- [16] T. G. Dietterich. Steps toward robust artificial intelligence. *AI Magazine*, 38(3):3–24, 2017.
- [17] B. C. Dostal. Enhancing situational understanding through employment of unmanned aerial vehicle. *Army Transformation Taking Shape: Interim Brigade Combat Team Newsletter*, 01-18, 2007.
- [18] D. Fierens, G. Van den Broeck, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Janssens, and L. De Raedt. Inference and learning in probabilistic logic programs using weighted Boolean formulas. *Theory and Practice of Logic Programming*, 15(03):358–401, May 2015.
- [19] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [20] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.
- [21] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420. Association for Computational Linguistics, 2016.
- [22] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE CVPR*, 2017.
- [23] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [25] Y. LeCun et al. Lenet-5, convolutional neural networks. URL: <http://yann.lecun.com/exdb/lenet>, page 20, 2015.
- [26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [27] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. L. Webber and N. J. Nilsson, editors, *Readings in Artificial Intelligence*, pages 431 – 450. Morgan Kaufmann, 1981.
- [28] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., 1997.
- [29] L. M. Kaplan and M. Ivanovska. Efficient belief propagation in second-order Bayesian networks for singly-connected graphs. *International Journal of Approximate Reasoning*, 93:132–152, Feb. 2018.
- [30] A. D. Preece, F. Cerutti, D. Braines, S. Chakraborty, and M. Srivastava. Cognitive computing for coalition situational understanding. In *IEEE Smart World Congress 2017 Workshop: DAIS 2017 - Workshop on Distributed Analytics Infrastructure and Algorithms for Multi-Organization Federations*, 2017.
- [31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [32] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [33] M. Richardson and P. Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.
- [34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.
- [35] T. Sato. A statistical learning method for logic programs with distribution semantics. In *Proceedings of the 12th International Conference on Logic Programming (ICLP-95)*, 1995.
- [36] H. Siegelmann and E. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132 – 150, 1995.
- [37] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, oct 2017.
- [38] N. Silver. *The signal and the noise: why so many predictions fail—but some don't*. Penguin, 2012.
- [39] B. C. Smith. *Reflection and semantics in a procedural language*. PhD thesis, Laboratory for Computer Science, MIT, 1982.
- [40] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, Jan. 2008.
- [41] C. D. Wickens, H. Li, A. Santamaria, A. Sebok, and N. B. Sarter. Stages and levels of automation: An integrated meta-analysis. *Human Factors*, 54(4):389–393, 2010.
- [42] E. Wright, S. Mahoney, K. Laskey, M. Takikawa, and T. Levitt. Multi-entity Bayesian networks for situation assessment. In *Proc. of the 5th Intl. Conf. on Information Fusion*, pages 804–811, 2002.
- [43] H. Zhuge and L. He. Automatic maintenance of category hierarchy. *Future Generation Computer Systems*, 67:1 – 12, 2017.